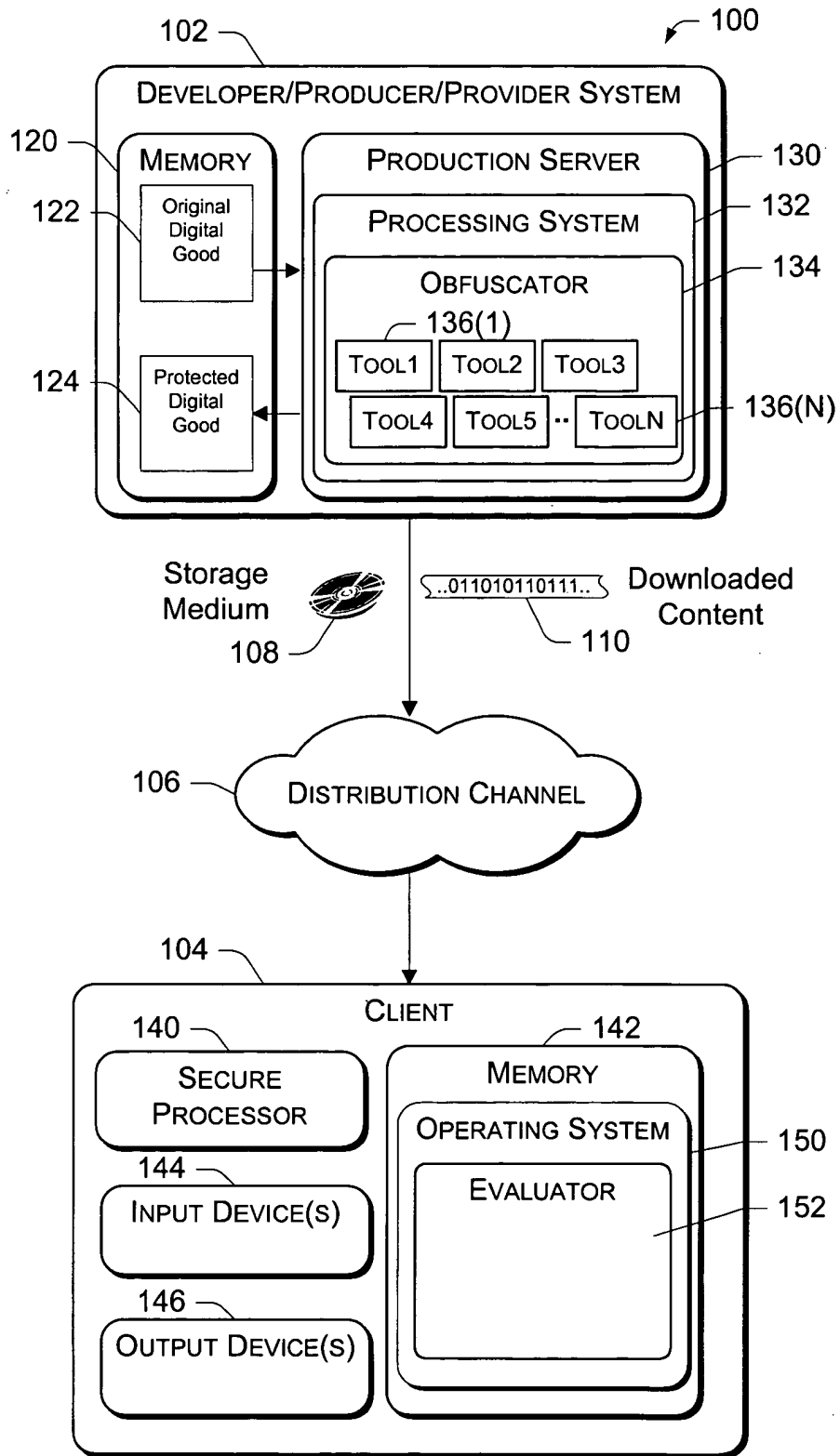
*Fig. 9*

*Fig. 1*

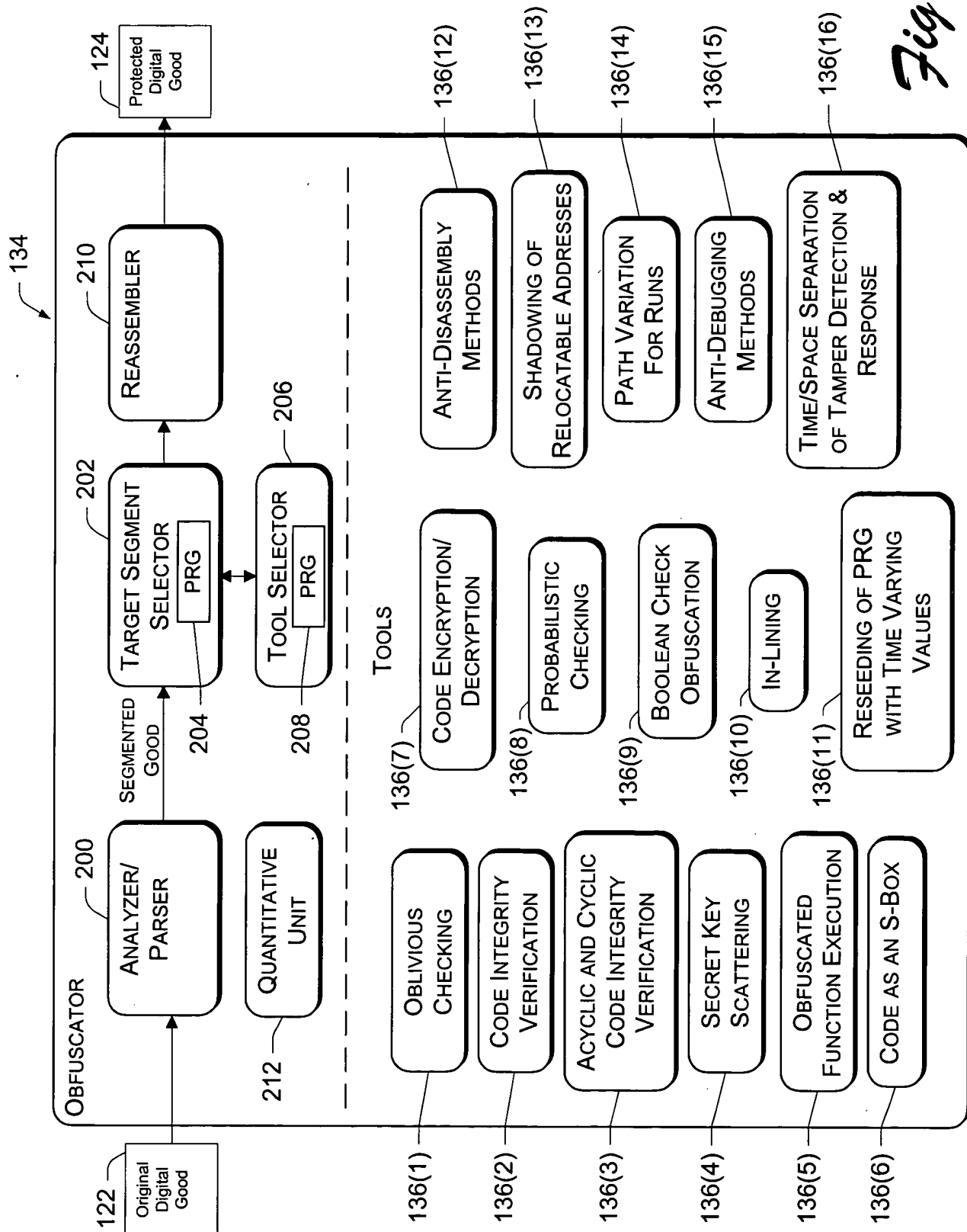


Fig. 2



Fig. 4

Diagram 124 illustrates a sequence of CP blocks (500(1) to 500(N)) connected by curved arrows, indicating a flow or sequence. The blocks are arranged in a circular pattern, with a break symbol (two wavy lines) indicating a continuation of the sequence. The blocks are labeled 500(1), 500(2), 500(3), ..., 500(N-2), 500(N-1), and 500(N). Each block contains the label "CP". The arrows show a sequence starting from 500(1) and proceeding through 500(2), 500(3), ..., 500(N-1), and finally 500(N), which then loops back to 500(1).

Fig. 5

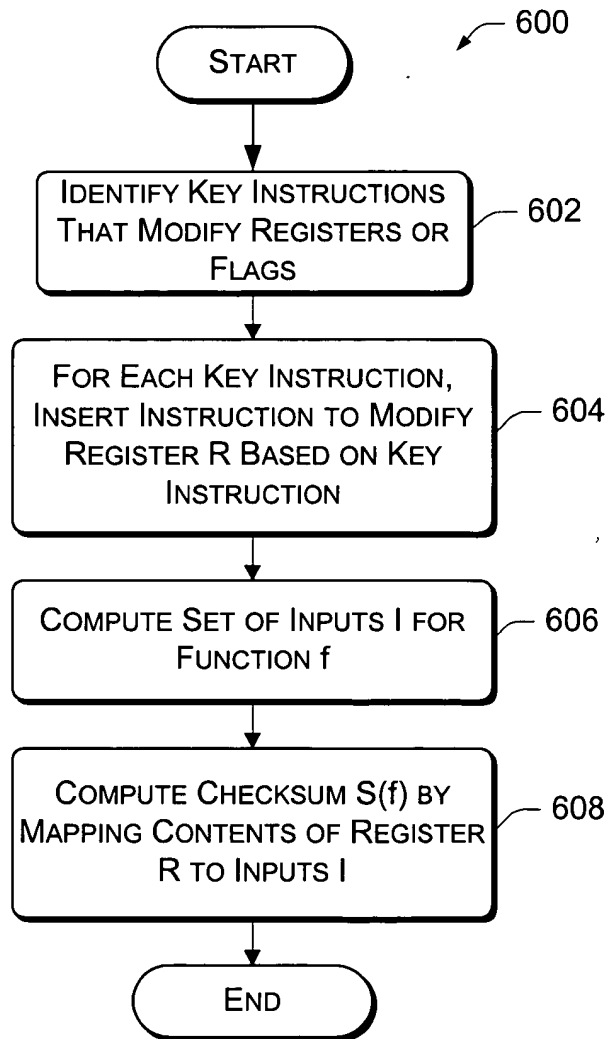
*Fig. 6*

Diagram 700 illustrates a memory layout for a segmented program. The diagram shows two segments, SEGMENT S1 and SEGMENT S2, stacked vertically. SEGMENT S1 is divided into two parts: a top part with diagonal hatching and a bottom part with horizontal dashed lines. SEGMENT S2 is divided into two parts: a top part with diagonal hatching and a bottom part with horizontal dashed lines. The diagram is bounded by a vertical line on the left and a vertical line on the right. The top of the diagram is labeled '700' with an arrow pointing to the top right corner. The bottom of the diagram is labeled 'Call Verify_Function(ID1), CS1 location' with an arrow pointing to the bottom left corner. The diagram includes labels for 'CS1', 'CP2', 'CS2', and 'CP1'. The vertical axis is labeled with 'BEGIN(ID1)', 'BEGIN(ID2)', 'END(ID1)', and 'END(ID2)'.

Fig. 7

Diagram 800 illustrates a method for determining a time slot for a second communication. The diagram shows a timeline with two segments, SEGMENT S3 and SEGMENT S4. SEGMENT S3 is bounded by BEGIN(ID3) and END(ID3). SEGMENT S4 is bounded by BEGIN(ID4) and END(ID4). A horizontal line separates the two segments. Below the line, a box labeled CP4 is shown. Above the line, a box labeled CP3 is shown. The diagram also shows a series of vertical lines representing time slots, with some slots marked with 'T'.

Fig. 8